

VISVESVARAYA TECHNOLOGICAL UNIVERSITY

“JnanaSangama”, Belgaum -590014, Karnataka.



LAB REPORT

on

OBJECT ORIENTED JAVA PROGRAMMING

Submitted by

SAM MANU JACOB (1BM23CSS291)

in partial fulfillment for the award of the degree of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING



B.M.S. COLLEGE OF ENGINEERING

(Autonomous Institution under VTU)

BENGALURU-560019 Sep

2024-Jan 2025

B. M. S. College of Engineering,
Bull Temple Road, Bangalore 560019
(Affiliated To Visvesvaraya Technological University, Belgaum)
Department of Computer Science and Engineering



CERTIFICATE

This is to certify that the Lab work entitled "**OBJECT ORIENTED JAVA PROGRAMMING**" carried out by **SAM MANU JACOB(1BM23CS291)**, who is bonafide student of **B. M. S. College of Engineering**. It is in partial fulfillment for the award of **Bachelor of Engineering in Computer Science and Engineering** of the Visvesvaraya Technological University, Belgaum during the year 2024-25. The Lab report has been approved as it satisfies the academic requirements in respect of **Object-Oriented Java Programming Lab - (23CS3PCOOJ)** work prescribed for the said degree.

Dr. Nandhini Vineeth

Associate Professor,
Department of CSE,
BMSCE, Bengaluru

Dr. Kavitha Sooda

Professor and Head,
Department of CSE
BMSCE, Bengaluru

INDEX

Sl. No.	Date	Experiment Title	Page No.
1			
2			
3			
4			
5			
6			
7			
8			
9			
10			

Include all the 8 programs as instructed in the classroom.

The order to be maintain for every program is

Question

Observation writeup images (complete)

Soft copy of the program

Screenshot of the output

**Editable copies of 9th and 10th program s are attached here.
Analyze and execute and include both in the lab record pdf
giving the same order as above. Explanation for both topics
are included in textbook.**

Include page numbers from this page onwards

LABORATORY PROGRAM – 1

Develop a Java program that prints all real solutions to the quadratic equation $ax^2 + bx + c = 0$. Read in a, b, c and use the quadratic formula. If the discriminant $b^2 - 4ac$ is negative, display a message stating that there are no real solutions.

DATE : 26/9/24

005 lab

1. Develop a Java Program that prints all real solutions to the quadratic equation $ax^2 + bx + c = 0$.
Read in a, b, c and use the quadratic formula. If the discriminant $b^2 - 4ac$ is negative, display a message stating that there are no real solutions.

```
import java.util.Scanner;
public class B.E
{
    public static void main (String args)
    {
        Scanner s = new Scanner (System.in);
        System.out.print ("Enter value of a ");
        double a = s.nextDouble();
        System.out.print ("Enter value of b ");
        double b = s.nextDouble();
        System.out.print ("Enter value of c ");
        double c = s.nextDouble();
        double d = b*b - 4*a*c;
        if (d > 0)
        {
            double r1 = (-b + Math.sqrt(d)) / (2*a);
            double r2 = (-b - Math.sqrt(d)) / (2*a);
            System.out.println ("The roots are real and distinct");
            System.out.println ("Roots are " + r1 + " and " + r2);
        }
        else if (d == 0)
        {
            double r = -b / (2*a);
            System.out.println ("Root has one real soln");
            System.out.println ("Root is " + r);
        }
        else
            System.out.println ("There are no real solutions");
    }
}
```

26/9/24

Output for program:	
Enter value of a : 1	Enter value of a : 1
Enter value of b : 2	Enter value of b : 2
Enter value of c : 3	Enter value of c : 1
There are no real solutions	Roots have one real solution Root is : -1
Enter value of a : 2	
Enter value of b : 6	
Enter value of c : 2	
Roots are real and distinct : -0.58196 and -2.618033	

```

import java.util.Scanner;
class QE
{
    public static void main(String xx[])
    {
        Scanner sc= new Scanner(System.in);
        System.out.println("Enter value a : ");
    }
}

```

```

double a = sc.nextDouble();
System.out.println("Enter value b :");
double b = sc.nextDouble();
System.out.println("Enter value c :");
double c = sc.nextDouble();

double d = b*b-4*a*c;
if(d>0)
{
    double r1= (-b+Math.sqrt(d))/(2*a);
    double r2= (-b-Math.sqrt(d))/(2*a);
    System.out.println("The roots are real and distinct");
    System.out.println("Roots are "+r1+" and "+r2);
}
else if(d==0)
{
    double r = -b/(2*a);
    System.out.println("Roots is "+r);
}
else
    System.out.println("There are no real solutions");

}

```

```

C:\Users\sammj\OneDrive\Desktop\JAVA LAB\lab 1>java QE
Enter value a :
1
Enter value b :
2
Enter value c :
3
There are no real solutions

C:\Users\sammj\OneDrive\Desktop\JAVA LAB\lab 1>java QE
Enter value a :
1
Enter value b :
2
Enter value c :
1
Roots is -1.0

C:\Users\sammj\OneDrive\Desktop\JAVA LAB\lab 1>java QE
Enter value a :
2
Enter value b :
6
Enter value c :
2
The roots are real and distinct
Roots are -0.3819660112501051 and -2.618033988749895

```

LABORATORY PROGRAM – 2

Develop a Java program to create a class Student with members usn, name, an array credits and an array marks. Include methods to accept and display details and a method to calculate SGPA of a student.

PAGE : 3 / 10 DATE : 31.10.2011

2. Develop a Java program to create a class student with members usn, name, an array credits and an array marks.
Include methods to accept and display details and a method to calculate SGPA of a student

```
import java.util.Scanner;
class Student {
    String usn;
    String name;
    int[] credits;
    int[] marks;
    public Student (int numSubjects) {
        credits = new int [numSubjects];
        marks = new int [numSubjects];
    }
    void acceptDetails () {
        Scanner s = new Scanner (System.in);
        System.out.print ("USN: ");
        usn = s.nextLine ();
        System.out.print ("Name: ");
        name = s.nextLine ();
        for (int i = 0, i < credits.length ; i++)
            System.out.println ("Enter credits for subject " +
                (i+1) + ": ");
        credits [i] = s.nextInt ();
        System.out.println ("Enter marks for subject " +
            (i+1) + ": ");
        marks [i] = s.nextInt ();
    }
}
```

void displayDetails()

{

 System.out.println("USN: " + USN);

 System.out.println("Name: " + name);

 for (int i = 0; i < credits.length; i++)

 {

 System.out.println("Subject" + (i + 1) + ": credits = " + credits[i] + ", marks = ", marks[i]);

 }

 double calculateSG()

{

 double totalCredits = 0;

 double totalPoints = 0;

 for (int i = 0; i < credits.length; i++)

 {

 totalCredits += credits[i];

 totalPoints += (marks[i] * 10.0) + credits[i];

 }

}

 if (totalCredits == 0)

 {

 return 0;

 }

 else

 {

 return (totalPoints / totalCredits);

 }

}

g

```
public class StudentSGPACalculator {  
    public static void main (String [] args) {  
        Scanner s = new Scanner (System.in);  
        System.out.println ("Enter no of subjects : ");  
        int numSubjects = s.nextInt();  
        Student st = new Student (numSubjects);  
        st.acceptDetails ();  
        st.displayDetails ();  
        double sgpa = st.calculateSGPA ();  
        System.out.println ("SGPA : " + sgpa);  
    }  
}
```

Output:

Enter no. of subjects: 3

USN: 1BM23CS291

Name: Sam

Enter credits for subject 1: 4

Enter marks for subject 1: 95

Enter credits for subject 2: 3

Enter marks for subject 2: 84

Enter credits for subject 3: 2

Enter marks for subject 3: 78

✓
3/10 | 29

USN: 1BM23CS291

Name: Sam

Subject 1: credits: 4 Marks: 95

Subject 2: credits: 3 Marks: 84

Subject 3: credits: 2 Marks: 78

SGPA: 9.222222

10, 9, 8

```
import java.util.Scanner;
class Student
{
    String usn;
    String name;
    int[] credits;
    int[] marks;
    Student(){}
    Student(int numSubjects)
    {
        credits = new int[numSubjects];
        marks = new int[numSubjects];
    }
    void acceptDetails()
    {
        Scanner sc= new Scanner(System.in);
        System.out.println("Enter USN : ");
        usn = sc.next();
        System.out.println("Enter Name : ");
        name = sc.next();

        for(int i=0; i < credits.length;i++)
        {
            System.out.println("Enter credits for the subject "+(i+1)+" :");
            credits[i]=sc.nextInt();
            System.out.println("Enter marks for subject "+(i+1)+" :");
            marks[i]=sc.nextInt();
        }
    }
    void displayDetails()
    {
        System.out.println("USN : "+usn);
        System.out.println("Name : "+name);
        for(int i=0;i<credits.length;i++)
        {
            System.out.println(" subject "+(i+1)+" : credits = "+credits[i]+" Marks = "+marks[i]);
        }
    }
}
```

```
double calculateSG()
{
    double totalCredits = 0;
    double totalPoints = 0;
    for( int i=0; i<credits.length; i++)
    {
        totalCredits += credits[i];
        if (marks[i]==100)
            totalPoints +=(10*credits[i]);
        else
            totalPoints +=((int)(marks[i]+10)/10)*credits[i];

    }
    if (totalCredits == 0)
        return 0;
    else
        return (totalPoints/totalCredits);
}
}

class StSgCalc
{
    public static void main( String[] xx)
    {
        Scanner s = new Scanner(System.in);
        System.out.println("Enter no. of subjects : ");
        int numSubjects= s.nextInt();
        Student st= new Student(numSubjects);
        st.acceptDetails();
        st.displayDetails();
        double sgpa = st.calculateSG();
        System.out.println("SGPA :" +sgpa);
    }
}
```

```
C:\Users\sammj\OneDrive\Desktop\JAVA LAB\Lab 2>java StSgCalc
Enter no. of subjects :
3
Enter USN :
1BM23CS291
Enter Name :
Sam
Enter credits for the subject 1 :
4
Enter marks for subject 1 :
95
Enter credits for the subject 2 :
3
Enter marks for subject 2 :
84
Enter credits for the subject 3 :
2
Enter marks for subject 3 :
78
USN : 1BM23CS291
Name : Sam
    subject 1 : credits = 4 Marks = 95
    subject 2 : credits = 3 Marks = 84
    subject 3 : credits = 2 Marks = 78
SGPA :9.22222222222221
```

LABORATORY PROGRAM – 3

Create a class Book which contains four members: name, author, price, num_pages. Include a constructor to set the values for the members. Include methods to set and get the details of the objects. Include a `toString()` method that could display the complete details of the book. Develop a Java program to create n book objects.

PAGE : 19 / 10 24 /

A Create a class Book which contains four members: name, author, price, num_pages. Include a constructor to set the values for the members. Include methods to set and get details of the objects. Include a `toString()` method that displays the complete details. Develop a java program to create a book object

```
class book {  
    String name;  
    String author;  
    double price;  
    int numPages;  
    public book (String name, String author, double price,  
    int numPages)  
    {  
        this.name = name;  
        this.author = author;  
        this.price = price;  
        this.numPages = numPages;  
    }  
    String getName()  
}
```

3

~~to print~~

~~String getName()~~

PAGE: _____
DATE: _____

```

void setDetails (String name, String author,
                 double price, int numPages)
{
    this.name = name;
    this.author = author;
    this.price = price;
    this.numPages = numPages;
}

void displayDetails () {
    System.out.println ("Book Name: " + name);
    System.out.println ("Book Author: " + author);
    System.out.println ("Book Price: " + price);
    System.out.println ("Book Pages: " + numPages);
}

public String toString {
    return (name + ";" + author + ";" + price + ";" + numPages);
}

```

class BookDemo

```

public static void main (String [] args) {
    Scanner sc = new Scanner (System.in);
    int n = 3;
    Book [] b = new Book [n];
    b[0] = new Book ("The Da Vinci Code", "Dan Brown",
                    1500, 350);
    b[1] = new Book ("Roots", "Alex Haley", 1650, 250);
    b[2] = new Book ("The Kite Runner", "Khalid Hosseini",
                    1580, 370);
}

```

for (int i = 0; i < n; i++)

b[i] = new Book ("Book Name", "Author", Price, NumPages);

```
for (sc.nextLine("Enter no of books: "));  
n = sc.nextInt();  
for (int i = 0; i < n; i++)  
{
```

String

```
System.out.print("Enter name: ")
```

```
String n = sc.nextLine();
```

```
System.out.print("Enter author: ")
```

```
String a = sc.nextLine();
```

```
int p
```

```
System.out.print("Enter price: ")
```

```
int p = sc.nextInt();
```

```
System.out.print("Enter no of pages: ")
```

```
int page = sc.nextInt();
```

```
{
```

```
b[i] = new Book(n, a, p, page);
```

```
b[i].displayDetails();
```

```
}
```

```
for (int i = 0; i < n; i++)
```

```
{ System.out.println("All book details:");
```

```
b[i].displayDetails(); System.out.println(b[i]);
```

```
}
```

~~done~~

~~int c~~

3

8

Output:

Enter no of books : 2

Enter name : The DA VINCI CODE

Enter Author : DAN BROWN

Enter price : 1200.00

Enter pages : 1600

Enter name : The Kite Runner

Enter Author : Khaled Hosseini

Enter price : 345.00

Enter no of pages : 371

All book details :

The DA VINCI CODE, DAN BROWN, 1200, 1600

The Kite Runner, Khaled Hosseini, 345, 371

```
import java.util.Scanner;
class Book
{
    String name;
    String author;
    double price;
    int numpages;

    Book(String name, String author, double price, int numpages)
    {
        this.name=name;
        this.author=author;
        this.price=price;
        this.numpages=numpages;
    }

    void setDetails(String name, String author, double price, int numPages)
    {
        this.name=name;
        this.author=author;
        this.price=price;
        this.numpages=numPages;
    }

    void displayDetails()
    {
        System.out.println("BookName:"+name);
        System.out.println("BookAuthor:"+author);
        System.out.println("BookPrice:"+price);
        System.out.println("BookPages:"+numpages);
    }

    public String toString()
    {
        return (name+ "," +author+ "," +price+"numpages");
    }
}

class BookDemo
{
    public static void main(String[] args)
```

```
{  
    Scanner sc = new Scanner(System.in);  
    System.out.println("Enter no of books:");  
    int n= sc.nextInt();  
    sc.nextLine();  
    Book[] b=new Book[n];  
    for(int i =0 ; i<n ;i++)  
    {  
        System.out.println("Enter name:");  
        String na = sc.nextLine();  
        System.out.println("Enter author:");  
        String a = sc.nextLine();  
        System.out.println("Enter price:");  
        int p = sc.nextInt();  
        System.out.println("Enter no of pages:");  
        int page = sc.nextInt();  
        sc.nextLine();  
  
        b[i]=new Book(na,a,p,page);  
        b[i].displayDetails();  
    }  
    System.out.println("All Book Details");  
    for (int i=0;i<n;i++)  
    {  
        System.out.println(b[i]);  
    }  
}
```

```
C:\Users\sammj\OneDrive\Desktop\JAVA LAB\Lab 3>java BookDemo
Enter no of books:
2
Enter name:
The Da Vinci Code
Enter author:
Dan Brown
Enter price:
350
Enter no of pages:
400
BookName:The Da Vinci Code
BookAuthor:Dan Brown
BookPrice:350.0
BookPages:400
Enter name:
The Kite Runner
Enter author:
KhaledHousseni
Enter price:
385
Enter no of pages:
390
BookName:The Kite Runner
BookAuthor:KhaledHousseni
BookPrice:385.0
BookPages:390
All Book Details
The Da Vinci Code,Dan Brown,350.0numpages
The Kite Runner,KhaledHousseni,385.0numpages
```

LABORATORY PROGRAM – 4

Develop a Java program to create an abstract class named Shape that contains two integers and an empty method named printArea(). Provide three classes named Rectangle, Triangle and Circle such that each one of the classes extends the class Shape. Each one of the classes contain only the method printArea() that prints the area of the given shape.

PAGE: / / DATE: / /

2) Develop a java program to create an abstract class named shape that contains two integers and an empty method named printArea(), provide three class name rectangle, triangle, and circle such that each one of the classes extends the class shape. Each one of the classes contains only one the method printArea()

```
import java.util.Scanner;  
  
abstract class shape  
{  
    int dim1;  
    int dim2;  
  
    public shape()  
    {  
        this.dim1 = dim1;  
        this.dim2 = dim2;  
    }  
  
    public abstract void printArea();  
}  
  
class Rectangle extends shape  
{  
    public Rectangle (int length, int width)  
    {  
        dim1 = length;  
        dim2 = width;  
    }  
  
    public void printArea()  
    {  
        int area = dim1 * dim2;  
        System.out.println(" Area of rectangle " + area);  
    }  
}
```

class triangle extends shape

{

public triangle (int base, int height)

{

dim1 = base;

dim2 = height;

}

public void printArea()

{

double area = 0.5 * dim1 * dim2;

System.out.println ("Area of triangle : " + area);

}

}

class circle extends shape

{

public circle (int radius)

{

dim1 = radius;

dim2 = 0;

}

public void printArea()

{

double area = Math.pi * dim1 * dim1;

System.out.println ("Area of circle : " + area);

}

}

class shapdemo

{

public static void main (String[] args)

{

Scanner sc = new Scanner (System.in);

System.out.println ("Enter length and breadth of rectangle");

int $a1$ = sc.nextInt();
int $a2$ = sc.nextInt();

shape rectangle = new shape();

rectangle $r1$ = new rectangle($a1$, $a2$);
 $r1.$ printArea();

System.out.println("Enter base and height of triangle");

int $d1$ = sc.nextInt();

$d2$ = sc.nextInt();

triangle $t1$ = new triangle($d1$, $d2$);
 $t1.$ printArea();

System.out.println("Enter radius");

$d1$ = sc.nextInt();

circle $c1$ = new circle($d1$);

$c1.$ printArea();

g

Output

Enter Length and Breadth of Rectangle

10 12

Area of Rectangl 120

Enter Base and Height of Triangle

12 24

Area of Triangle 144.0

Enter radius of Circle

12

Area of Circle 452.304

N
23/12/21

```
import java.util.Scanner;

abstract class Shape {
    int dim1;
    int dim2;

    public Shape(int dim1, int dim2) {
        this.dim1 = dim1;
        this.dim2 = dim2;
    }

    public abstract void printArea();
}

class Rectangle extends Shape {
    public Rectangle(int length, int width) {
        super(length, width);
    }

    public void printArea() {
        int area = dim1 * dim2;
        System.out.println("Area of rectangle: " + area);
    }
}

class Triangle extends Shape {
    public Triangle(int base, int height) {
        super(base, height);
    }

    public void printArea() {
        double area = 0.5 * dim1 * dim2;
        System.out.println("Area of triangle: " + area);
    }
}

class Circle extends Shape {
    public Circle(int radius) {
        super(radius, 0);
    }
}
```

```

public void printArea() {
    double area = Math.PI * dim1 * dim1;
    System.out.println("Area of circle: " + area);
}

class ShapeDemo {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter length and breadth of rectangle:");

        int d1;
        d1 = sc.nextInt();
        int d2;
        d2 = sc.nextInt();

        Rectangle r1 = new Rectangle(d1, d2);
        r1.printArea();

        System.out.println("Enter base and height of triangle");
        d1 = sc.nextInt();
        d2 = sc.nextInt();
        Triangle t1 = new Triangle(d1, d2);
        t1.printArea();

        System.out.println("Enter radius");
        d1 = sc.nextInt();
        Circle c1 = new Circle(d1);
        c1.printArea();
    }
}

```

```

C:\Users\sammj\OneDrive\Desktop\JAVA LAB\lab 4>java ShapeDemo
Enter length and breadth of rectangle:
10 20
Area of rectangle: 200
Enter base and height of triangle
10 25
Area of triangle: 125.0
Enter radius
5
Area of circle: 78.53981633974483

```

LABORATORY PROGRAM – 5

Develop a Java program to create a class Bank that maintains two kinds of account for its customers, one called savings account and the other current account. The savings account provides compound interest and withdrawal facilities but no cheque book facility. The current account provides cheque book facility but no interest. Current account holders should also maintain a minimum balance and if the balance falls below this level, a service charge is imposed. Create a class Account that stores customer name, account number and type of account. From this derive the classes Cur-acct and Sav-acct to make them more specific to their requirements.

Include the necessary methods in order to achieve the following tasks:

- a) Accept deposit from customer and update the balance.
- b) Display the balance.
- c) Compute and deposit interest
- d) Permit withdrawal and update the balance Check for the minimum balance, impose penalty if necessary and update the balance.

5) Develop a Java Program to create a class Bank that maintains two kinds of accounts for its customers, one called savings account and another called current account. The savings account provides compound interest and withdrawal but no cheque book facility. The current account provides cheque book facility but no interest. Current account holders should also maintain a minimum balance and if the balance falls below this level, a service charge is imposed. Create a class Account that stores customer name Acc No. From this derive the class Cur Acc and Sav Acc to make them more specific to their requirements.

Include necessary methods:

- a) Accept deposit from customer and update the balance
- b) Display the balance
- c) Compute and deposit interest
- d) Permit withdrawal and update balance. Check for minimum balance and impose penalty.

import java.util.Scanner;

abstract class Account

{

String CustomerName;

int AccNo;

String AccType;

double balance;

Account (String customername, int AccNo, String AccType)

{

This. CustomerName = customername;

This. AccNo = accno;

This. AccType = accType;

double balance = 0;

{

PAGE : / /
DATE : / /

infost java

```
abstract void computeInterest();  
{  
    void deposit (double amount)  
    {  
        balance = balance + amount  
    }  
    void withdraw (double amount)  
    {  
        if (amount <= balance)  
        {  
            balance = balance - amount;  
        }  
        else  
        {  
            System.out.println ("Insufficient funds");  
        }  
    }  
    void displayBalance()  
    {  
        System.out.println ("Balance = " + balance);  
    }  
}
```

class Current extends Account

```
{  
    double minbalance = 2000;  
    int penalty = 100;  
    Current (String customerName, int AceNo)  
    {  
        super (CustomerName, Ace-No, "Current Account")  
    }  
    void computeInterest()  
    {  
        System.out.println ("No interest on Current Acc");  
    }  
}
```

void withdraw(double amount)
{

super.withdraw(amount);
if (balance < minbalance)
{

balance = balance - penalty;

System.out.println("Penalty of Rs 100 applied");

}

}

class SavAcc extends Account

{

double interest = 12;

SavAcc (String CustomerName, int Ac_No)

{

super(CustomerName, Ac_no, "Savings Account");

}

void computeInterest()

{

double interestamt = 12.0 / 100 * balance;

balance = balance + interestamt;

}

void withdraw (double amount)

{

super.withdraw(amount);

}

class Bank

{

public static void main (String [] args)

{

Scanner sc = new Scanner (System.in);

Account c1 = new CustAcc ("Sam", 1);

Account s1 = new SavAcc ("Sam", 2);

while (true)

{

System.out.println ("Enter 1 for current account,
2 for savings account, 3 to exit");

int c = sc.nextInt();

if (c == 1)

{

while (true)

{

System.out.println ("Enter 1 to deposit,
2 to withdraw,

3 to display balance,

4 to calculate interest");

int op = sc.nextInt();

if (op == 1)

{

System.out.println ("Enter amount");

double amt = sc.nextDouble();

c1.deposit(amt);

3

else if (op == 2)

{

System.out.println ("Enter amount");

double amt = sc.nextDouble();

c1.withdraw(amt);

g

else if ($op == 3$):

{

c1. displayBalance();

}

else if ($op == 4$):

{

c1. computeInterest();

}

else

{ break;

}

}

{

else if ($c == 2$):

{

while (true)

{

System.out.println("Enter 1 to Deposit, 2 to withdraw,
3 to display and 4 to compute");

int op = sc.nextInt();

if ($op == 1$):

{

System.out.println("Enter amount");

double amount = sc.nextDouble();

s1. deposit(amount);

}

else if ($op == 2$):

{

System.out.println("Enter amount");

double amount = sc.nextDouble();

s1. withdraw(amount);

else if ($op == 3$):

{

DATE: / /

st. displayBalance()

g
else if (ob == 0)
{

st. combtBalance();

g

else

break;

g

else

break;

g

g

Output:

Fahr 1 for current Account, 2 for saving Account,
1 for withdraw, 3 to exit

Fahr 1 to deposit, 2 to withdraw, 3 to display balance, 4 to
comptBalance

1

Enter amount: 1000

Fahr 1 to deposit, 2 to withdraw, 3 to display balance, 4 to benefit
3

Balance > 1000.0

Fahr 1 to ---

2

Fahr Amount:

4000

Penalty of Rs 100 applied

Fahr 1 to deposit, 2 to withdraw...
4

PAGE: / / /
DATE: / / /

No interest on Current Account
 Enter 1 to deposit
 5
 Enter 1 to deposit for Savings Acc, 2 for Current Acc, 3 to withdraw.
 2
 Enter 1 to Deposit, 2 to Withdraw, 3 to DisplayBalance &
 to calculate Interest
 1
 Enter Amount: 5000
 Enter 1 to deposit, 2 to withdraw
 2
 Enter Amount:
 4000
 Enter 1 to deposit, 2 to withdraw
 3
 Balance = 500
 Enter 1 to deposit, 2 to withdraw...
 4
 Enter 1 to deposit, 2 to withdraw, 3 to Display, 4 to calculate
 3
 Balance = 160

~~N
23/2/21~~

```

import java.util.Scanner;

abstract class Account
{
    String CustomerName;
    int Acc_no;
    String AccType;
    double balance;

    Account(String CustomerName, int Acc_no, String AccType)
    {
        this.CustomerName=CustomerName;
        this.Acc_no= Acc_no;
        this.AccType = AccType;
    }
}

```

```
balance=0;
}

abstract void computeInterest();

void deposit(double amount)
{
    balance = balance+amount;
}

void withdraw(double amount)
{
    if(amount<=balance)
    {
        balance = balance - amount;
    }
    else
    {
        System.out.println("Oops! Insufficient Funds!");
    }
}

void displayBalance()
{
    System.out.println("Balance = "+balance);
}

class CurrAcc extends Account
{
    double minbalance=2000;
    int penalty=100;

    CurrAcc(String CustomerName, int Acc_no)
    {
        super(CustomerName,Acc_no,"Current Account");
    }

    void computeInterest()
    {
        System.out.println("No interest on current account");
    }

    void withdraw(double amount)
```

```

{
    super.withdraw(amount);

    if (balance<minbalance)
    {
        balance= balance-penalty;
        System.out.println("Penalty of rs 100 applied");
    }
}

}

class SavAcc extends Account
{
    double interest=12;

    SavAcc(String CustomerName, int Acc_no)
    {
        super(CustomerName,Acc_no,"Savings Account");
    }

    void computeInterest()
    {
        double interestamt = 12.0/100.0*balance;
        balance = balance+interestamt;
    }

    void withdraw(double amount)
    {
        super.withdraw(amount);
    }
}

class Bank
{
    public static void main(String[] args)
    {
        Scanner sc = new Scanner(System.in);

        Account c1= new CurrAcc("Sam",1);
        Account s1= new SavAcc("Sam",2);

        while(true)
        {
            System.out.println("Enter 1 for current account , 2 for savings account, 3 to

```

```
exit");
    int c= sc.nextInt();
    if (c==1)
    {
        while(true)
        {
            System.out.println("Enter 1 to Deposit, 2 to Withdraw, 3 to
DisplayBalance, 4 to                               computeInterest");
            int op = sc.nextInt();

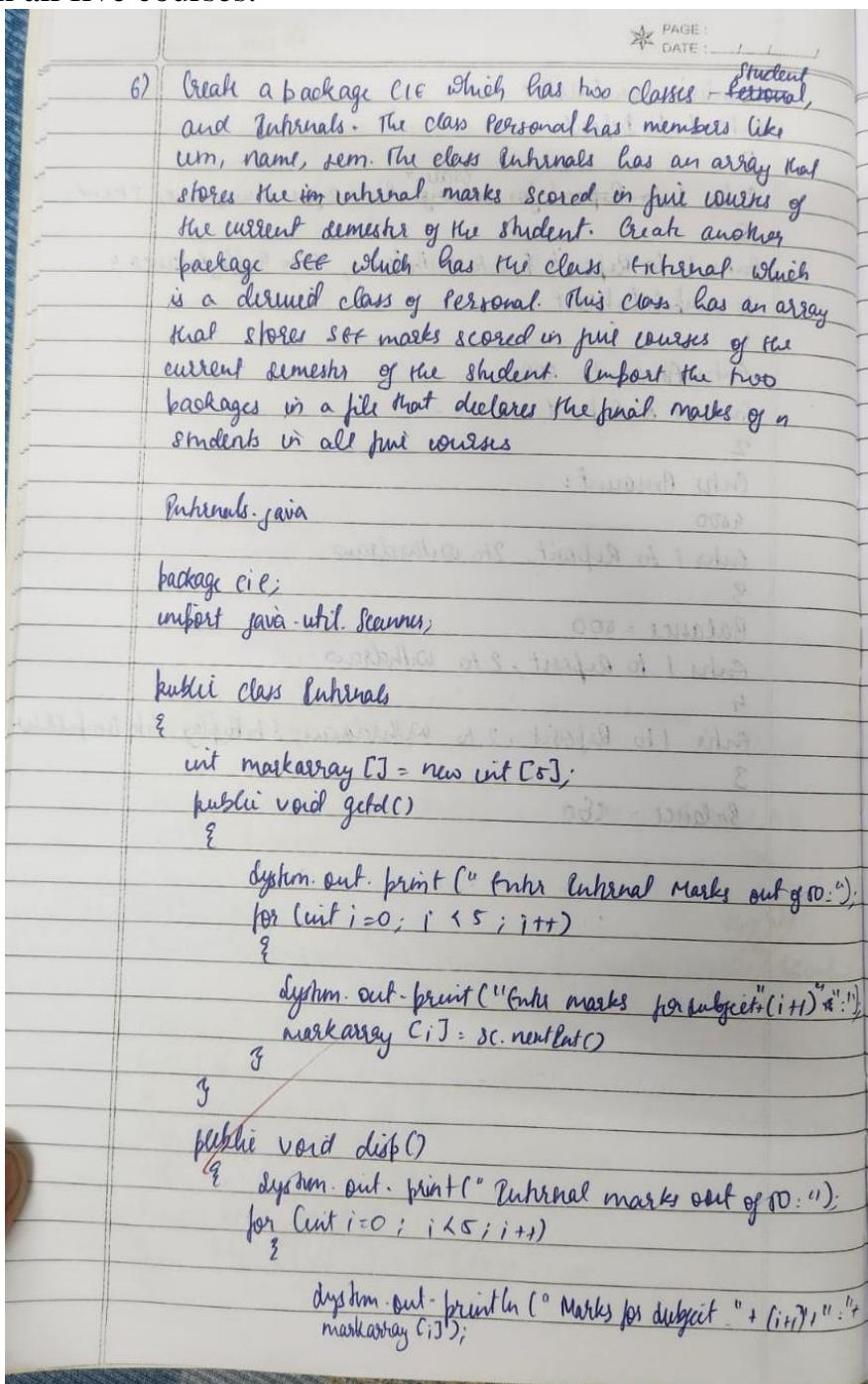
            if(op==1)
            {
                System.out.println("Enter amount:");
                double amnt = sc.nextDouble();
                c1.deposit(amnt);
            }
            else if(op==2)
            {
                System.out.println("Enter amount:");
                double amnt = sc.nextDouble();
                c1.withdraw(amnt);
            }
            else if(op==3)
            {
                c1.displayBalance();
            }
            else if(op==4)
            {
                c1.computeInterest();
            }
            else
                break;
        }
    }
    else if(c==2)
    {
        while(true)
        {
            System.out.println("Enter 1 to Deposit, 2 to Withdraw, 3 to
DisplayBalance, 4 to                               computeInterest");
            int op = sc.nextInt();
            if(op==1)
            {
                System.out.println("Enter amount:");
```

```
        double amnt = sc.nextDouble();
        s1.deposit(amnt);
    }
    else if(op==2)
    {
        System.out.println("Enter amount:");
        double amnt = sc.nextDouble();
        s1.withdraw(amnt);
    }
    else if(op==3)
    {
        s1.displayBalance();
    }
    else if(op==4)
    {
        s1.computeInterest();
    }
    else
        break;
}
}
else
    break;
}
}
```

```
Enter 1 for current account , 2 for savings account, 3 to exit  
1  
Enter 1 to Deposit, 2 to Withdraw, 3 to DisplayBalance, 4 to  
1  
Enter amount:  
5000  
Enter 1 to Deposit, 2 to Withdraw, 3 to DisplayBalance, 4 to  
3  
Balance = 5000.0  
Enter 1 to Deposit, 2 to Withdraw, 3 to DisplayBalance, 4 to  
2  
Enter amount:  
4500  
Penalty of rs 100 applied  
Enter 1 to Deposit, 2 to Withdraw, 3 to DisplayBalance, 4 to  
4  
No interest on current account  
Enter 1 to Deposit, 2 to Withdraw, 3 to DisplayBalance, 4 to  
5  
Enter 1 for current account , 2 for savings account, 3 to exit  
2  
Enter 1 to Deposit, 2 to Withdraw, 3 to DisplayBalance, 4 to  
1  
Enter amount:  
5000  
Enter 1 to Deposit, 2 to Withdraw, 3 to DisplayBalance, 4 to  
2  
Enter amount:  
4500  
Enter 1 to Deposit, 2 to Withdraw, 3 to DisplayBalance, 4 to  
3  
Balance = 500.0  
Enter 1 to Deposit, 2 to Withdraw, 3 to DisplayBalance, 4 to  
4  
Enter 1 to Deposit, 2 to Withdraw, 3 to DisplayBalance, 4 to  
3  
Balance = 560.0  
Enter 1 to Deposit, 2 to Withdraw, 3 to DisplayBalance, 4 to
```

LABORATORY PROGRAM – 6

Create a package CIE which has two classes - Personal and Internals. The class Personal has members like usn, name, sem. The class Internals has an array that stores the internal marks scored in five courses of the current semester of the student. Create another package SEE which has the class External which is a derived class of Personal. This class has an array that stores the SEE marks scored in five courses of the current semester of the student. Import the two packages in a file that declares the final marks of n students in all five courses.



System.out.println("n");

g
g
g

Student.java:

package cie;
import java.util.Scanner;

public class Student

{

String usn;

String name;

int dem;

int marks[] = new int [5];

public void getdata()

{

Scanner sc = new Scanner(System.in);

System.out.print("Enter USN: ");

usn = sc.nextInt();

System.out.print("Enter Name: ");

name = sc.next();

System.out.print("Enter Dem: ");

dem = sc.nextInt();

g

public void disp()

{

System.out.println("USN: " + usn);

System.out.println("Name: " + name);

System.out.println("Dem: " + dem);

g

g

Finals.java

package see;

import java.util.Scanner;
import eie.Student;

public class Finals extends Student

{

int markarray [] = new int [5];

public Finals ()

{

super();

{

public void getd()

{

super.getd();

{

public void solif()

{

super.dif();

{

public void getd()

{

Scanner sc = new Scanner (System.in);

for (int i = 0; i < 5; i++)

{

System.out.print ("Enter marks for subject " + (i+1) + ": ");

markarray [i] = sc.nextInt();

System.out.print ("\n");

{

public void dif()

{

for (int i = 0; i < 5; i++)

{

System.out.println ("Marks for subject " + (i+1), " " + markarray [i]);

{

public void disp()

{

Scanner sc = new Scanner (System.in);
System.out.print("Enter marks out of 10 : ");
for (int i=0; i<5; i++)

{

System.out.println (" Marks for subject " + (i+1) + " : " +
marksarray[i]);

System.out.print("In ");

{

System.out.println();

Dem0.java

import java.util.Scanner;

import com.EI.Student;

import com.EI.Parents;

import com.EI.Father;

class Dem0

{ static int n

static Parents m[] = new Parents[n];

static Parents e[] = new Parents [n];

public static void dispDetails ()

{ for (int i=0; i<n; i++)

{ e[i].sdif();

em[i].disp();

g e[i].clif();

}

' public static void compute ()

{

for (int i=0; i<n; i++)

{

e[i] = new Internals();

i[n] = new Internals();

e[i].setDC();

i[n].getDC();

e[i].getDC();

for (int j=0; j<5; j++)

{

e[i].Jmarks[j] = e[i].markarray[j] +

i[n].markarray[j];

}

public static void main (String args)

{

Scanner sc = new Scanner (System.in);

System.out.println ("Enter no. of students: ");

n = sc.nextInt();

compute ();

System.out.println ("Displaying all student details: ")
dispdetails();

Output:

Enter Number of Students:

2

Enter USN: 1

Enter Name: a

Enter DC: 1

Enter Internal Marks Out of 100: Enter marks for subject 1: 35

n. Write a program that demonstrates handling of exceptions

Entire marks for subject 2: 36

Entire marks for subject 3: 37

Entire marks for subject 4: 38

Entire marks for subject 5: 39

Entire internal Marks out of 50: Entire marks for subject 1: 41

Entire marks for subject 2: 42

Entire marks for subject 3: 43

Entire marks for subject 4: 44

Entire marks for subject 5: 45

Entire USN: 2

Entire Name: S

Entire Sem: 2

Entire External Marks out of 100: Entire marks for subject 1: 27

Entire marks for subject 2: 28

Entire marks for subject 3: 29

Entire marks for subject 4: 30

Entire marks for subject 5: 31

Entire Internal Marks out of 50: Entire marks for subject 1: 32

Displaying all 8th

2: 33

3: 34

4: 35

5: 36

Displaying all student details:

USN: 1

Name: A

Sem: 1

Marks for subject 1: 76

2: 77

3: 80

4: 83

5: 83

UAN: 2

Name: b

Dem: 2

External Marks out

Marks for subject 1: 59

Mark for subject 2: 61

3 : 63

4 : 65

5 : 67

✓
23/12/21

Internals.java

```
package cie;
import java.util.Scanner;

public class Internals
{
    Scanner sc = new Scanner(System.in);
    public int markarray[] = new int[5];

    public void getd()
    {
        System.out.print("Enter Internal Marks out of 50 : ");
        for(int i=0;i<5;i++)
        {
            System.out.print("Enter Marks for subject "+(i+1)+" : ");
            markarray[i] = sc.nextInt();
            System.out.print("\n");
        }
    }

    public void disp()
    {
        System.out.print("Internal Marks out of 50 : ");
        for(int i=0;i<5;i++)
        {
            System.out.println("Marks for subject "+(i+1)+" : " +markarray[i]);
            System.out.print("\n");
        }
    }
}
```

Student.java

```
package cie;
import java.util.Scanner;

public class Student
{
    String usn;
    String name;
    int sem;
    public int finals[] = new int[5];
    public void getd()
    {
        Scanner sc = new Scanner (System.in);
        System.out.print("Enter USN:");
        usn = sc.next();
        System.out.print("Enter Name:");
        name = sc.next();
        System.out.print("Enter Sem:");
        sem = sc.nextInt();
    }

    public void disp()
    {
        System.out.println("USN:"+usn);

        System.out.println("Name:"+name);

        System.out.println("Sem:"+sem);

    }
}
```

```
Externals.java
package see;
import java.util.Scanner;
import cie.Student;

public class Externals extends Student
{
    Scanner sc = new Scanner(System.in);
    public int markarray[] = new int[5];
    public int finals[] = new int[5];
    public Externals()
    {
        super();
    }

    public void sgetd()
    {
        super.getd();
    }
    public void sdisp()
    {
        super.disp();
    }
    public void getd()
    {
        Scanner sc = new Scanner (System.in);
        for(int i=0;i<5;i++)
        {
            System.out.print("Enter Marks for subject "+(i+1)+" : ");
            markarray[i] = sc.nextInt();
            System.out.print("\n");
        }
    }
    public void disp()
    {
        Scanner sc = new Scanner (System.in);
        System.out.print("Total Marks out of 50 : ");
        for(int i=0;i<5;i++)
        {
            System.out.println("Marks for subject "+(i+1)+" : " +finals[i]);
            System.out.print("\n");
        }
    }
}
```

```
Demo.java
import java.util.Scanner;
import cie.Student;
import cie.Internals;
import see.Externals;

class Demo
{
    static int n;
    static Internals[] in;
    static Externals[] e;

    public static void dispdetails()
    {
        for (int i = 0; i < n; i++)
        {
            e[i].sdisp();
            in[i].disp();
            e[i].disp();
        }
    }

    public static void compute()
    {
        for (int i=0;i<n;i++)
        {
            e[i]= new Externals();
            in[i]= new Internals();
            e[i].sgetd();
            in[i].getd();
            e[i].getd();
            for (int j=0; j<5;j++)
            {
                e[i].finals[j]=e[i].markarray[j]+in[i].markarray[j];
            }
        }
    }

    public static void main(String args[])
    {
        Scanner sc = new Scanner(System.in);
```

```
System.out.println("Enter no of students: ");
n=sc.nextInt();
in = new Internals[n];
e = new Externals[n];
compute();
System.out.println("Displaying all Student details: ");
dispdetails();

}

}
```

```
Enter no of students:
2
Enter USN:1
Enter Name:a
Enter Sem:1
Enter Internal Marks out of 50 : Enter Marks for subject 1 : 35

Enter Marks for subject 2 : 36

Enter Marks for subject 3 : 37

Enter Marks for subject 4 : 38

Enter Marks for subject 5 : 39

Enter Marks for subject 1 : 41

Enter Marks for subject 2 : 42

Enter Marks for subject 3 : 43

Enter Marks for subject 4 : 44

Enter Marks for subject 5 : 45

Enter USN:2
Enter Name:b
Enter Sem:2
Enter Internal Marks out of 50 : Enter Marks for subject 1 : 27

Enter Marks for subject 2 : 28

Enter Marks for subject 3 : 29

Enter Marks for subject 4 : 30

Enter Marks for subject 5 : 31

Enter Marks for subject 1 : 32

Enter Marks for subject 2 : 33
```

```
Enter Marks for subject 3 : 34
Enter Marks for subject 4 : 35
Enter Marks for subject 5 : 36
Displaying all Student details:
USN:1
Name:a
Sem:1
Internal Marks out of 50 : Marks for subject 1 : 35
Marks for subject 2 : 36
Marks for subject 3 : 37
Marks for subject 4 : 38
Marks for subject 5 : 39
Total Marks out of 50 : Marks for subject 1 : 76
Marks for subject 2 : 78
Marks for subject 3 : 80
Marks for subject 4 : 82
Marks for subject 5 : 84
USN:2
Name:b
Sem:2
Internal Marks out of 50 : Marks for subject 1 : 27
Marks for subject 2 : 28
Marks for subject 3 : 29
Marks for subject 4 : 30
Marks for subject 5 : 31
Total Marks out of 50 : Marks for subject 1 : 59
Marks for subject 2 : 61
Marks for subject 3 : 63
Marks for subject 4 : 65
Marks for subject 5 : 67
```

```
C:\Users\sammj\OneDrive\Desktop\JAVA LAB\lab 6>
```

LABORATORY PROGRAM – 7

Write a program that demonstrates handling of exceptions in inheritance tree. Create a base class called “Father” and derived class called “Son” which extends the base class. In Father class, implement a constructor which takes the age and throws the exception WrongAge() when the input age=father’s age.

7. Write a program that demonstrates handling of exceptions in inheritance tree. Create a base class called “Father” and derived class called “Son”, which extends the base class. In Father class, implement a constructor, which takes the age and throws the exception WrongAge() when the input age < 0. In Son class, implement a constructor that uses both father and sons age and throws exception if sons age > fathers age.

→ class WrongAgeException extends Exception

{
↳ public WrongAgeException (String message)

}
duper (message);

{
↳ class Father

{
int age;

public Father (int age) extends WrongAgeException
{
↳ if (age < 0)

}
throw new WrongAgeException ("Age cannot be negative");

{
↳ this.age = age;

{
↳ class Son extends Father

{
int sonAge;

public Son (int fatherAge, int sonAge) throws
WrongAgeException
{
↳ if (sonAge < 0)

}
throw new WrongAgeException ("Age cannot be negative");

{
↳ if (sonAge >= fatherAge)
throw new WrongAgeException ("Son's age
cannot be greater than father's age");

{
↳ this.sonAge = sonAge;

public class Demo {

public static void main (String [] args)

{ try

Scanner sc = new Scanner (System . in);
System.out.println ("Enter father's age : ");

int age = sc.nextInt ();

Father father = new Father (age);

try

System.out.println ("Enter son's age : ");

Son son = new Son (age);

int sage = sc.nextInt ();

Son son = new Son (age, sage);

g catch (WrongAgeException e)

g System.out.println (" Exception caught ! ", e.getMessage ())

g }

g }

Output :

Enter father's age : -10

Age exception caught : WrongAgeException: Father's age cannot be -ve

Enter father's age : 10

Enter son's age : -10

Age exception caught : WrongAgeException: Son's age cannot be -ve

✓ 23/12/29

Enter father's age : 10

Enter son's age : 60

Age exception caught : WrongAgeException: Son's age cannot be greater than father's

```
import java.util.Scanner;

class WrongAgeException extends Exception {
    public WrongAgeException(String message) {
        super(message);
    }
}

class Father {
    int age;
    public Father(int age) throws WrongAgeException {
        if (age < 0) {
            throw new WrongAgeException("Age cannot be negative");
        }
        this.age = age;
    }
}

class Son extends Father {
    int sonAge;
    public Son(int fatherAge, int sonAge) throws WrongAgeException {
        super(fatherAge);
        if (sonAge < 0) {
            throw new WrongAgeException("Age cannot be negative");
        }
        if (sonAge >= fatherAge) {
            throw new WrongAgeException("Son's age cannot be greater than father's
age");
        }
        this.sonAge = sonAge;
    }
}

class Demo {
    public static void main(String[] args) {
        try {
            Scanner sc = new Scanner(System.in);
            System.out.println("Enter father's age: ");
            int fage = sc.nextInt();
            Father father = new Father(fage);

            System.out.println("Enter son's age: ");
            int sage = sc.nextInt();
            Son son = new Son(father.age, sage);
        }
    }
}
```

```
        System.out.println("Father's age: " + father.age);
        System.out.println("Son's age: " + son.sonAge);
        System.out.println();
    } catch (WrongAgeException e) {
        System.out.println("Exception caught: " + e.getMessage());
    }
}
}
```

```
C:\Users\sammj\OneDrive\Desktop\JAVA LAB\lab 7>java Demo
Enter father's age:
-40
Exception caught: Age cannot be negative

C:\Users\sammj\OneDrive\Desktop\JAVA LAB\lab 7>java Demo
Enter father's age:
40
Enter son's age:
-20
Exception caught: Age cannot be negative

C:\Users\sammj\OneDrive\Desktop\JAVA LAB\lab 7>java Demo
Enter father's age:
40
Enter son's age:
50
Exception caught: Son's age cannot be greater than father's age

C:\Users\sammj\OneDrive\Desktop\JAVA LAB\lab 7>
```

LABORATORY PROGRAM – 8

Write a program which creates two threads, one thread displaying “BMS College of Engineering” once every ten seconds and another displaying “CSE” once every two seconds.

PAGE : / /
DATE : / /

8) Write a program which creates two threads, one thread displaying “BMS College of Engineering” once every 10 seconds and another displaying “CSE” once every two seconds.

```
class BMS implements Runnable  
{  
    Thread t;  
    BMS(){}  
    t = new Thread (this, "BMS Thread");  
    System.out.println("Child Thread: " + t);  
    public void run()  
    {  
        try  
        {  
            while (true)  
            {  
                System.out.println("BMS College of Engineering");  
                Thread.sleep(10000);  
            }  
        }  
        catch (InterruptedException e)  
        {  
            System.out.println("Child Interrupted");  
        }  
    }  
}
```

class CSE implements Runnable

{

 Thread t;

 CSE()

{

 t = new Thread(this, "CSE_Thread");

}

 public void run()

{

 try

{

 for(;;) (true),

{

 System.out.println("CSE");

{ Thread.sleep(2000);

{

}

 catch (InterruptedException e)

{

 System.out.println("And Interrupted");

{

}

 class ThreadDemo

{

 public static void main(String[] args)

{

 BMS t1 = new BMS();

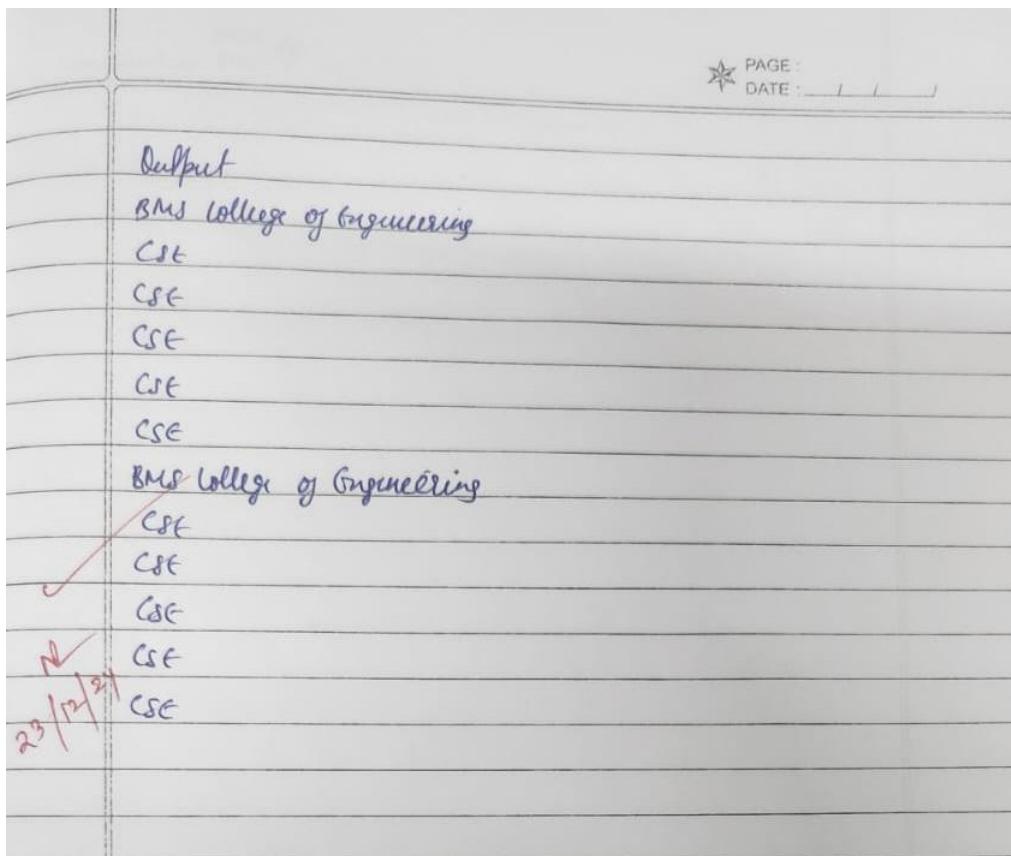
 CSE t2 = new CSE();

 t1.start();

 t2.start();

{

}



```
class BMS implements Runnable {  
    Thread t;  
    BMS() {  
        t = new Thread(this, "BMS Thread");  
        System.out.println("Child thread: " + t);  
    }  
  
    public void run() {  
        try {  
            while (true) {  
                System.out.println("BMS College of Engineering");  
                Thread.sleep(10000);  
            }  
        } catch (InterruptedException e) {  
            System.out.println("Child interrupted");  
        }  
    }  
}
```

```
class CSE implements Runnable {  
    Thread t;  
    CSE() {  
        t = new Thread(this, "CSE Thread");  
    }  
  
    public void run() {  
        try {  
            while (true) {  
                System.out.println("CSE");  
                Thread.sleep(2000);  
            }  
        } catch (InterruptedException e) {  
            System.out.println("Child interrupted");  
        }  
    }  
}  
  
class ThreadDemo {  
    public static void main(String[] args) {  
        BMS t1 = new BMS();  
        CSE t2 = new CSE();  
        t1.t.start();  
        t2.t.start();  
    }  
}
```

```
C:\Users\sammj\OneDrive\Desktop\JAVA LAB\lab 8>java ThreadDemo
Child thread: Thread[#30, BMS Thread, 5, main]
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
```

LABORATORY PROGRAM - 9

Write a program that creates a user interface to perform integer divisions. The user enters two numbers in the text fields, Num1 and Num2. The division of Num1 and Num2 is displayed in the Result field when the Divide button is clicked. If Num1 or Num2 were not an integer, the program would throw a NumberFormatException. If Num2 were Zero, the program would throw an ArithmeticException. Display the exception in a message dialog box.

9.

Write a program that creates a user interface to perform integer division. The user enters two numbers in the text fields, Num1 and Num2. The division of Num1 and Num2 is displayed in the Result field. When the Divide button is clicked. If Num1 & were zero, the program would throw an Arithmetic exception. Display the exception in a message dialog box.

```
import java.awt.*;  
import java.awt.event.*;
```

```
public class DivisionMain1 extends Frame implements ActionListener  
{
```

```
    JTextField num1, num2;  
    JButton dResult;  
    JLabel outResult;  
    String out = " ";  
    double resultNum;  
    int flag = 0;
```

```
    public DivisionMain1()  
{
```

```
        setLayout(new FlowLayout());  
        dResult = new JButton("RESULT");  
        label1 number1 = new Label("Number 1:",  
        label.RIGHT);  
        label number2 = new Label("Number 2:",  
        label.RIGHT);  
        num1 = new TextField(5);  
        num2 = new TextField(5);  
        outResult = new Label("Result:", label.RIGHT);
```

```
add(number1);  
add(num1);  
add(number2);  
add(num2);  
add(dResult);  
add(dResult);  
num1.addActionListener(this);  
num2.addActionListener(this);  
dResult.addActionListener(this);  
addWindowListener(new WindowAdapter())  
{  
    public void windowClosing(WindowEvent we)  
    {  
        System.exit(0);  
    }  
}
```

```
public void actionPerformed(ActionEvent ae)  
{  
    int n1, n2;  
    try  
    {  
        if (ae.getSource() == dResult)  
        {  
            n1 = Integer.parseInt(num1.getText());  
            n2 = Integer.parseInt(num2.getText());  
            if (n2 == 0)  
                throw new ArithmeticException();  
            out = n1 + n2 + " ";  
            resultNum = n1 / n2;  
            out += String.valueOf(resultNum);  
            repaint();  
        }  
    }  
}
```

n1 = Integer.parseInt(num1.getText());
n2 = Integer.parseInt(num2.getText());

if (n2 == 0)

throw new ArithmeticException();

out = n1 + n2 + " ";

resultNum = n1 / n2;

out += String.valueOf(resultNum);

repaint();

catch (NumberFormatException e1)

{

flag = 1;

out = "Number Format Exception!" + e1;

repaint();

g

catch (ArithmaticException e2)

{

flag = 1;

out = "Quotient by 0 Exception!" + e2;

repaint();

g

public void paint(Graphics g)

{

if (flag == 0)

{

g.drawString(out, outResult.getX() + outResult.get

width(), outResult.getY() +

outResult.getHeight() - 8);

else

g.drawString(out, 10, 10, 200);

flag = 0;

g

public static void main (String[] args)

{

DimensionMain1 frame = new DimensionMain1();

frame.setSize (300, 200);

frame.setTitle ("Dimension App");

frame.setVisible (true);

g

Output

Number 1: 24

Number 8

Result

Result: 24 8 3.0

```
import
java.awt.*;
import
java.awt.event.*;

public class DivisionMain1 extends Frame implements ActionListener
{
    TextField
    num1,num2;
    Button dResult;
    Label
    outResult;
    String
    out="";
    double
    resultNum;
    int flag=0;

    public DivisionMain1()
    {
        setLayout(new FlowLayout());

        dResult = new Button("RESULT");
        Label number1 = new Label("Number
1:",Label.RIGHT); Label number2 = new
Label("Number      2:",Label.RIGHT);
        num1=new TextField(5);
        num2=new TextField(5);
        outResult = new Label('Result:',Label.RIGHT);

        add(number1
);
        add(num1);
        add(number2
);
        add(num2);
        add(dResult);
        add(outResul
```

```
t);

num1.addActionListener(this);
num2.addActionListener(this);
dResult.addActionListener(this);
addWindowListener(new
WindowAdapter()
{
    public void windowClosing(WindowEvent we)
    {
        System.exit(0);
    }
});

public void actionPerformed(ActionEvent ae)
{
    int
    n1,n2;
    try
    {
        if (ae.getSource() == dResult)
        {
            n1=Integer.parseInt(num1.getText());
            n2=Integer.parseInt(num2.getText());

            /*if(n2==0)
             * throw new
             * ArithmeticException();*/
            out=n1+"
            +n2+" ";
            resultNum=n1/n2;
            out+=String.valueOf(result
            Num); repaint();

        }
    }
    catch(NumberFormatException e1)
    {
        flag=1;
        out="Number Format Exception!
        +e1; repaint();
    }
}
```

```
        }
    catch(ArithmeticException e2)
    {
        flag=1;
        out="Divide by 0 Exception!
"+e2; repaint();
    }

}
public void paint(Graphics g)
{
    if(flag==0)
        g.drawString(out,outResult.getX()+outResult.getWidth(),outResult.getY()+outResult.getHeight()-8);
    else
        g.drawString(out,10
0,200); flag=0;
}
public static void main(String[] args)
{
    DivisionMain1 dm=new
DivisionMain1(); dm.setSize(new
Dimension(800,400));
dm.setTitle("DivisionOfIntegers");
dm.setVisible(true);
}
}
```

OUTPUT

```
D:\NotePad++\Java>javac DivisionMain1.java
D:\NotePad++\Java>java DivisionMain1

```

LABORATORY PROGRAM - 10

Demonstrate Interprocess communication and deadlock

Demonstrate interprocess communication and deadlock

class A

{

int n;

boolean valuedet = false;

synchronized int get()

{

while (!valuedet)

{

try {

System.out.println ("In consumer waiting (" + n + ")");

wait();

}

catch (InterruptedException e)

{

System.out.println ("Interrupted exception caught");

}

System.out.println ("Got : " + n);

valuedet = true;

System.out.println ("In infinite producer (" + n + ")");

notify();

return n;

}

}

synchronized void put (int n)

{

while (valuedet)

{

try {

System.out.println ("In producer waiting (" + n + ")");

wait();

}

catch (UnknownHostException) {

System.out.println("UnknownHostException caught");

This.n = n;

valuedet = true;

System.out.println("Put " + n);

System.out.println("In Intmain consumer(" + n + ");
notify();

}

class Producer implements Runnable

{

Q q;

Producer(Q q)

{

This.q = q;

new Thread(this, "Producer").start();

}

public void run()

{

int i = 0;

while (i < 15)

{

q.put(i++);

}

}

{

class Consumer implements Runnable

{

Q

Consumer(Q q)

{

this.q = q;

new Thread(this, "Consumer").start();

S

public void run()

{

int i=0;

while(i<10)

{

int x=q.get();

System.out.println("Consumed "+x);

i++;

S

{

}

class Producer

{

public static void main(String args[])

{

Q q = new Q();

new Producer(q),

new Consumer(q),

System.out.println("Press Control-C to stop");

S

{

```
class Q {
int n;
boolean valueSet = false;

synchronized int get() {
while(!valueSet)
try {
System.out.println("\nConsumer waiting\n");
wait();
} catch(InterruptedException e) {
System.out.println("InterruptedException caught");
}
System.out.println("Got: " + n);
valueSet = false;
System.out.println("\nIntimate Producer\n");
notify();
return n;
}

synchronized void put(int n) {
while(valueSet)
try {
System.out.println("\nProducer waiting\n");
wait();
} catch(InterruptedException e) {
System.out.println("InterruptedException caught");
}
this.n = n;
valueSet = true;
System.out.println("Put: " + n);
System.out.println("\nIntimate Consumer\n");
notify();
}
}

class Producer implements Runnable {
Q q;
Producer(Q q) {
this.q = q;
new Thread(this, "Producer").start();
}
public void run() {
int i = 0;
```

```
while(i<15) {
q.put(i++);
}
}
}

class Consumer implements Runnable {
Q q;
Consumer(Q q) {
this.q = q;
new Thread(this, "Consumer").start();
}
public void run() {
    int i=0;
while(i<15) {
int r=q.get();
System.out.println("consumed:"+r);
i++;
}
}
}

class PCFixed {
public static void main(String args[]) {
Q q = new Q();
new Producer(q);
new Consumer(q);
System.out.println("Press Control-C to stop.");
}
}
```

```
MainThread entered A.foo
RacingThread entered B.bar
MainThread trying to call B.last()
RacingThread trying to call A.last()
```

ii. Demonstration of deadlock

PAGE: _____
DATE: _____

Demonstration of Deadlock

class A

{

synchronized void foo(B b)

{

String name = Thread.currentThread().getName();

System.out.println("[" + name + " entered A:foo");

try

{

Thread.sleep(1000);

}

catch (Exception e)

{

System.out.println("A InterruptedException");

}

synchronized void last() {

{

System.out.println("[" + name + " exited A");

}

}

class B

{

synchronized void bar(A a)

{

String name = Thread.currentThread().getName();

System.out.println("[" + name + " entered B:bar");

try { Thread.sleep(1000); }

catch (Exception e) {

System.out.println("B InterruptedException");

}

System.out.println("[" + name + " trying to call A:last");

a.last();

}

synchronized void es last() {

{

System.out.println("[" + name + " exited A:last");

}

class Readlock implements Runnable,

{

 a = new A(),

 b = new B();

 Readlock()

{

 Thread.currentThread().setName("Main Thread");

 Thread t = new Thread(this, "RaceThread");

 t.start(); a.foo(b);

 System.out.println("Back in main thread")

}

 public void run()

{

 b.bar(a);

 System.out.println("Back in the thread");

}

 public static void main(String args[])

{

 new Readlock();

}

} //

```

class A
{
    synchronized void foo(B b)
    { String name = Thread.currentThread().getName();
        System.out.println(name + " entered A.foo");
        try { Thread.sleep(1000); }
        catch(Exception e) { System.out.println("A Interrupted"); }
        System.out.println(name + " trying to call B.last()"); b.last(); }
        synchronized void last() { System.out.println("Inside A.last"); }
    }

class B {
    synchronized void bar(A a) {
        String name = Thread.currentThread().getName();
        System.out.println(name + " entered B.bar");
        try { Thread.sleep(1000); }
        catch(Exception e) { System.out.println("B Interrupted"); }
        System.out.println(name + " trying to call A.last()"); a.last(); }
    synchronized void last() { System.out.println("Inside A.last"); }

}

class Deadlock implements Runnable
{
    A a = new A(); B b = new B();
    Deadlock() {
        Thread.currentThread().setName("MainThread");
        Thread t = new Thread(this, "RacingThread");
        t.start(); a.foo(b); // get lock on a in this thread.
        System.out.println("Back in main thread");
    }
    public void run() { b.bar(a); // get lock on b in other thread.
        System.out.println("Back in other thread");
    }
    public static void main(String args[]) { new Deadlock(); }
}

```

```
PS C:\Users\sammj\New folder (2)> javac DivisionMain1.java
PS C:\Users\sammj\New folder (2)> java PCFixed
Press Control-C to stop.
Put: 0

Intimate Consumer

Producer waiting

Got: 0

Intimate Producer

Put: 1

Intimate Consumer

Producer waiting

consumed:0
Got: 1

Intimate Producer

consumed:1
Put: 2

Intimate Consumer

Producer waiting

Got: 2

Intimate Producer
```

```
Producer waiting

Got: 13

Intimate Producer

consumed:13
Put: 14

Intimate Consumer

Got: 14

Intimate Producer

consumed:14
PS C:\Users\sammj\New folder (2)> |
```
